

Note

- Instructions have been included for each segment. You do not have to follow them exactly, but they are included to help you think through the steps.

```
In [4]: # Dependencies and Setup

import pandas as pd
import numpy as np

# File to Load (Load file as per path)
file_to_load = "Resources/purchase_data.csv"

# Read Purchasing File and store into Pandas data frame
purchase_data = pd.read_csv(file_to_load)
purchase_data.head()
```

Out[4]:

	Purchase ID	SN	Age	Gender	Item ID	Item Name	Price
0	0	Lisim78	20	Male	108	Extraction, Quickblade Of Trembling Hands	3.53
1	1	Lisovynya38	40	Male	143	Frenzied Scimitar	1.56
2	2	lthergue48	24	Male	92	Final Critic	4.88
3	3	Chamassasya86	24	Male	100	Blindscythe	3.27
4	4	Iskosia90	23	Male	131	Fury	1.44

Player Count

- Display the total number of players

```
In [5]: numberofplayers=purchase_data['SN'].value_counts()

players_df = pd.DataFrame({
    "Total Players": [len(numberofplayers)]
})
players_df
```

Out[5]:

Total Players
0
576

Purchasing Analysis (Total)

- Run basic calculations to obtain number of unique items, average price, etc.
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

```
In [6]: #Take Price Column and Sum
Total_Revenue=purchase_data["Price"].sum()
#print(f"Total Revenue :",Total_Revenue)

#Take Price Column and find Mean
Avg_Purchase_Price=purchase_data["Price"].mean()
#print(f"Average Purchase Price :",round(Avg_Purchase_Price,2))

#Take Price Column and find total count
Total_Number_Purchase=purchase_data["Price"].count()
#print(f"Total Number of Purchase :", Total_Number_Purchase)

# Find number of Unique Value
Number_Unique_Item=purchase_data["Item Name"].nunique()
#print(f"Number of Unique Item :", Number_Unique_Item)

# Create DataFrame Using above variables

purchasing_df = pd.DataFrame({
    "Number of Unique Items": [Number_Unique_Item],
    "Average Price $"       : [round(Avg_Purchase_Price,2)],
    "Number of Purchases"   : [Total_Number_Purchase],
    "Total Revenue $"       : [Total_Revenue]
})
purchasing_df
```

Out[6]:

	Number of Unique Items	Average Price \$	Number of Purchases	Total Revenue \$
0	179	3.05	780	2379.77

- Percentage and Count of Male Players
- Percentage and Count of Female Players
- Percentage and Count of Other / Non-Disclosed

```
In [7]: # #Group by Gender & SN, find unique values

gender_count = purchase_data.groupby('Gender')['SN'].nunique()
Total_Gender_Count = purchase_data.groupby('Gender')['SN'].nunique().sum()

percentage_gender = (gender_count/Total_Gender_Count)*100

#print(percentage_gender)

gender_demo = pd.DataFrame({
    "Total Counts": gender_count,
    "Percentage of Players": round(percentage_gender,2),
})

# Index Added
# gender_demo.index = (["Female", "Male", "Other / Non-Disclosed"])
gender_demo.sort_values('Total Counts',ascending = False)
```

Out[7]:

	Total Counts	Percentage of Players
Gender		
Male	484	84.03
Female	81	14.06
Other / Non-Disclosed	11	1.91

Purchasing Analysis (Gender)

- Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. by gender
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

```

In [8]: # Group by Gender
gender_group_purchase_data = purchase_data.groupby(["Gender"])
gender_group_purchase_data.count()

purchase_count=gender_group_purchase_data["SN"].count()
#print(purchase_count)

avg_purchase_price=gender_group_purchase_data["Price"].mean()
#print(avg_purchase_price)

total_purchase_value=gender_group_purchase_data["Price"].sum()
#print(total_purchase_value)

avg_purchase_price_perperson = round((gender_group_purchase_data["Price"].sum() / gender_count),2)
#print(avg_purchase_price_perperson)

# Create new DataFrame
purchase_analysis_gender = pd.DataFrame({ "Purchase Count"           : purchase_count,
                                           "Average Purchase Price"    : round(avg_purchase_price,2),
                                           "Total Purchase Value"      : total_purchase_value,
                                           "Avg Total Purchase per Person": avg_purchase_price_perperson})

purchase_analysis_gender

```

Out[8]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
Gender				
Female	113	3.20	361.94	4.47
Male	652	3.02	1967.64	4.07
Other / Non-Disclosed	15	3.35	50.19	4.56

Age Demographics

- Establish bins for ages
- Categorize the existing players using the age bins. Hint: use `pd.cut()`
- Calculate the numbers and percentages by age group
- Create a summary data frame to hold the results
- Optional: round the percentage column to two decimal points
- Display Age Demographics Table

```
In [9]: age_bins = [0, 9.99, 14.99, 19.99, 24.99, 29.99, 34.99, 39.99, 100]
group_names = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]

purchase_data["Age_Summary"] = pd.cut(purchase_data["Age"], age_bins, labels=group_names)
purchase_data

purchase_data.head()
```

Out[9]:

	Purchase ID	SN	Age	Gender	Item ID	Item Name	Price	Age_Summary
0	0	Lisim78	20	Male	108	Extraction, Quickblade Of Trembling Hands	3.53	20-24
1	1	Lisovynya38	40	Male	143	Frenzied Scimitar	1.56	40+
2	2	lthergue48	24	Male	92	Final Critic	4.88	20-24
3	3	Chamassasya86	24	Male	100	Blindscythe	3.27	20-24
4	4	Iskosia90	23	Male	131	Fury	1.44	20-24

```

In [10]: group_data = purchase_data.groupby(['SN', 'Age_Summary'])['Age'].count()
         #print(group_data)

         total_counts = group_data.groupby('Age_Summary').count()
         total_counts

         # %Calculations
         percentage = round((total_counts/len(group_data))*100,2)
         percentage
         age_demographics = pd.DataFrame ({
                                     "Total Counts"           : total_counts,
                                     "Percentage of Players"  : percentage,
                                     })
         age_demographics

```

Out[10]:

	Total Counts	Percentage of Players
Age_Summary		
<10	17	2.95
10-14	22	3.82
15-19	107	18.58
20-24	258	44.79
25-29	77	13.37
30-34	52	9.03
35-39	31	5.38
40+	12	2.08

Purchasing Analysis (Age)

- Bin the purchase_data data frame by age
- Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. in the table below
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

```
In [11]: # Count purchase data using SN by age bin
purchase_data_item = purchase_data.groupby('Age_Summary')['SN'].count()
#print(purchase_data_item)
average_purchase_price = purchase_data.groupby('Age_Summary')['Price'].mean()
#print(round(average_purchase_price,2))
total_purchase_value = purchase_data.groupby('Age_Summary')['Price'].sum()
#print(total_purchase_value)
avg_total_purchase_per_person = total_purchase_value/total_counts
#print(round(avg_total_purchase_per_person,2))

purchase_analysis_df = pd.DataFrame ({ "Purchase Count": purchase_data_item,
                                       "Average Purchase Price ": round(average_purchase_price,2),
                                       "Total Purchase Value": total_purchase_value,
                                       "Avg Total Purchase per Person": round(avg_total_purchase_per_person,2),
                                       })
purchase_analysis_df
```

Out[11]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
Age_Summary				
<10	23	3.35	77.13	4.54
10-14	28	2.96	82.78	3.76
15-19	136	3.04	412.89	3.86
20-24	365	3.05	1114.06	4.32
25-29	101	2.90	293.00	3.81
30-34	73	2.93	214.00	4.12
35-39	41	3.60	147.67	4.76
40+	13	2.94	38.24	3.19

Top Spenders

- Run basic calculations to obtain the results in the table below
- Create a summary data frame to hold the results
- Sort the total purchase value column in descending order
- Optional: give the displayed data cleaner formatting
- Display a preview of the summary data frame

```

In [12]: top_spenders= purchase_data['SN'].value_counts()
          #print(top_spenders)

          average_purchase_price_spenders = purchase_data.groupby('SN')['Price'].mean()
          #print(round(average_purchase_price_spenders,2))

          total_purchase_value_spenders = purchase_data.groupby('SN')['Price'].sum()
          #print(round(total_purchase_value_spenders,2))

          # Cretaing DataFrame
          spender_analysis_df = pd.DataFrame ({ "Purchase Count"      : top_spenders,
                                                  "Average Purchase Price" : round(average
          _purchase_price_spenders,2),
                                                  "Total Purchase Value"   : total_purchas
          e_value_spenders,
                                                  })
          spender_analysis_df.sort_values('Purchase Count', ascending = False).head()

```

Out[12]:

	Purchase Count	Average Purchase Price	Total Purchase Value
Lisosia93	5	3.79	18.96
Iral74	4	3.40	13.62
Idastidru52	4	3.86	15.45
Asur53	3	2.48	7.44
Inguron55	3	3.70	11.11

Most Popular Items

- Retrieve the Item ID, Item Name, and Item Price columns
- Group by Item ID and Item Name. Perform calculations to obtain purchase count, item price, and total purchase value
- Create a summary data frame to hold the results
- Sort the purchase count column in descending order
- Optional: give the displayed data cleaner formatting
- Display a preview of the summary data frame


```
In [13]: #Group by Item ID & Item Name than perform calculations

most_popular_items= purchase_data.groupby('Item ID')['Item Name'].value_counts
()
#print(most_popular_items)

most_popular_items_price= purchase_data.groupby(['Item ID','Item Name'])['Price']
.mean()
#print(most_popular_items_price)

most_popular_items_price_total= purchase_data.groupby(['Item ID','Item Name'])
['Price'].sum()
#print(most_popular_items_price_total)

# Cretaing DataFrame

most_popular_items_df = pd.DataFrame ({ "Purchase Count"      : most_popular
_items,
                                     "Item Price"             : most_popular
_items_price,
                                     "Total Purchase Value"    : most_popular
_items_price_total,
                                     })
most_popular_items_df.sort_values('Purchase Count', ascending = False).head()
```

Out[13]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	Item Name			
178	Oathbreaker, Last Hope of the Breaking Storm	12	4.23	50.76
145	Fiery Glass Crusader	9	4.58	41.22
108	Extraction, Quickblade Of Trembling Hands	9	3.53	31.77
82	Nirvana	9	4.90	44.10
19	Pursuit, Cudgel of Necromancy	8	1.02	8.16

Most Profitable Items

- Sort the above table by total purchase value in descending order
- Optional: give the displayed data cleaner formatting
- Display a preview of the data frame

```
In [14]: # Above DataFrame in Ascending Order
most_popular_items_df.sort_values('Purchase Count', ascending = False).head()
```

Out[14]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	Item Name			
178	Oathbreaker, Last Hope of the Breaking Storm	12	4.23	50.76
145	Fiery Glass Crusader	9	4.58	41.22
108	Extraction, Quickblade Of Trembling Hands	9	3.53	31.77
82	Nirvana	9	4.90	44.10
19	Pursuit, Cudgel of Necromancy	8	1.02	8.16